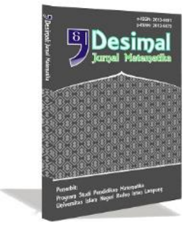
Contents lists available at DJM

DESIMAL: JURNAL MATEMATIKA

p-ISSN: 2613-9073 (print), e-ISSN: 2613-9081 (online), DOI 10.24042/djm

http://ejournal.radenintan.ac.id/index.php/desimal/index



Digital signature scheme with matrix-based approach

Syafrul Irawadi*, Hidayat Febiansyah, Maxrizal

Institut Sains Dan Bisnis Atma Luhur, Pangkalpinang

ARTICLE INFO

Article History

Received : 27-08-2024
Revised : 29-09-2024
Accepted : 29-10-2024
Published : 06-11-2024

Keywords:
Digital Signature; Matrix Scheme ; Matrix Signature.

*Correspondence: E-mail: syafrul@atmaluhur.ac.id

Doi:
10.24042/djm.v7i3.23956

ABSTRACT

The use of digital signatures in various electronic services such as e-transactions, e-commerce, and e-learning is necessary for today's humans. All types of these services are highly dependent on the privacy, integrity, and authenticity between the sender and recipient of the data. Mathematically, many digital signature schemes such as Rivest Shamir Adleman (RSA), Elgamal, and Elliptic Curve Cryptography (ECC) are made using the concept of integer multiplication. Previous research introduced the RSA signature with a square matrix that changes data as a matrix instead of integers. The security of the scheme depends on the matrix with order h. *The larger* h *the digit chosen, the better the level of protection. This modification makes this digital signature system more secure than systems using integers because the randomization process is more random and complicated. However, the operating system involves matrix exponentiation, requiring a lot of computing time and space. In this study, researchers changed the matrix exponentiation to ordinary matrix multiplication. The advantage is that the proposed algorithm has a faster computing speed because it only involves ordinary matrix multiplication. In the first step, the researcher forms several rectangular matrices as random variables for the key generation algorithm. Next, the researcher models the signing and signature verification algorithms. After that, the researcher codes in Mathematica and simulates the proposed signature scheme. In the final stage, the researcher performs a mathematical attack test analysis on the algorithm. The results show that the proposed scheme can generate keys and sign and verify signatures well. In addition, the proposed scheme system has also been tested for possible mathematical attacks.*

http://ejournal.radenintan.ac.id/index.php/desimal/index

INTRODUCTION

Various electronic services, such as e-transactions, e-commerce, and e-learning, are currently used to meet human needs ranging from business to personal activities. All of these types of services are highly dependent on aspects of privacy, integrity, and authenticity of data between the sender and recipient. For this reason, using digital signatures plays an essential role in ensuring the integrity of data transactions.

Digital signature schemes such as the Rivest Shamir Adleman (RSA) algorithm (Anshori, Dodu, & Wedananta, 2019), Elgamal (Ismail & Misro, 2022; Lalem, Laouid, Kara, Al-Khalidi, & Eleyan, 2023) and Elliptic Curve Cryptography (ECC) (Saepulrohman & Negara, 2021) use integer keys and involve integer powers. Some modifications of RSA and ElGamal with matrices (Gupta & Sanghi, 2021; Maxrizal & Irawadi, 2019; Raj & Sridhar, 2021) or variations of ElGamal (Fuchsbauer, Plouviez, & Seurin, 2020; Qin & Zhang, 2023; Saputra & Purnomo, 2018) can be used as relatively efficient and secure digital signature schemes. Researchers also provide comprehensive comparative studies on digital signatures of several RSA and Elgamal schemes (Swain, Pradhan, & Moharana, 2022).

Research by Gupta & Sanghi (2021) shows a modification of the RSA signature with the concept of a square matrix $n \times n$. The goal is to change the data to a matrix instead of integers to reduce the key size. The researcher proposed modifying the RSA digital signature scheme using a matrix with the order of h in the research. The larger the digit chosen of h, the better the level of security, namely randomization for the data matrix M^h. Note that this modification makes this digital signature system more secure than the system using integers because the randomization process is more random and complicated. This means that the operating system involves the matrix's power over integers, requiring a lot of computing time and space.

This study proposes an improvement to the study by Gupta & Sanghi (2021) by using ordinary matrices without having to use matrix exponentiation. The advantage is that the proposed algorithm has a faster computing speed because it only involves ordinary matrix multiplication. This is different from the study by Gupta & Sanghi (2021), which uses matrix exponentiation. Because both use the matrix concept, the proposed system maintains the same level of data security as the previous study.

METHOD

This research will modify the RSA digital signature system based on the matrix in previous research. If research by Gupta & Sanghi (2021) uses matrix exponentiation, then the proposed system only uses ordinary matrix multiplication. In the first step, the researcher will form several rectangular matrices as random variables for the key generation algorithm. Next, the signing and signature verification algorithms will be modeled. After that, the researcher will code in Mathematica and simulate the proposed signature scheme. To ensure the security of the scheme, the researcher will conduct a mathematical attack test analysis on the algorithm. The research flowchart of this study can be seen in Figure 1.

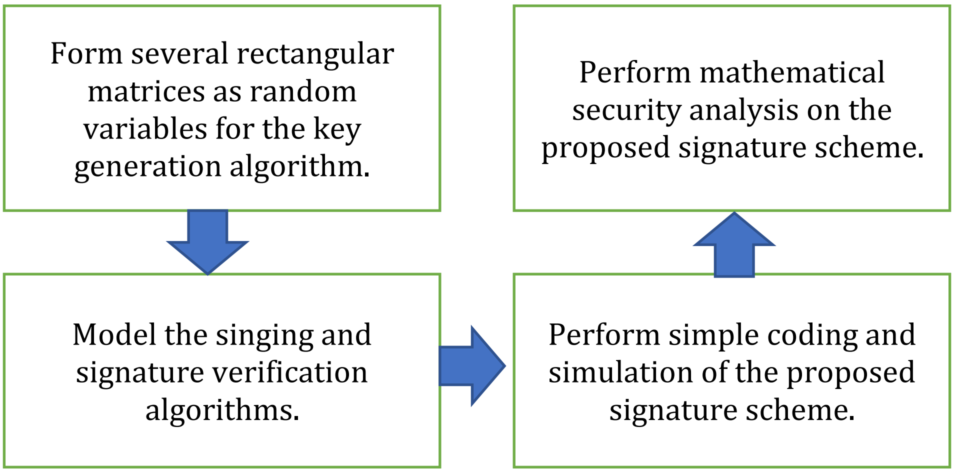

Form several rectangular matrices as random variables for the key generation algorithm.
Model the singing and signature verification algorithms.
Perform simple coding and simulation of the proposed signature scheme.
Perform mathematical security analysis on the proposed signature scheme.

Figure 1. Research Flowchart

RESULTS AND DISCUSSION

Background to the Development of the Proposed Signature Scheme

Let us consider a message matrix M and a nonsingular matrix A. Note that $M_{r\times n} = M_{r\times n}\left(A_{n\times n}\right)^{-1} A_{n\times n}$ holds. Next, we form $A_{n\times n} = B_{n\times m}C_{m\times n}$. If $m > n > r$, then

$$M_{r\times n} = M_{r\times n}\left(A_{n\times n}\right)^{-1} A_{n\times n}$$

$$M_{r\times n} = M_{r\times n}\left(A_{n\times n}\right)^{-1} B_{n\times m}C_{m\times n}$$

$$M_{r\times n} = M_{r\times n}\left[\left(A_{n\times n}\right)^{-1} B_{n\times m}\right]C_{m\times n}$$

From the last equation we assume $T_{r\times m} = M_{r\times n}\left[\left(A_{n\times n}\right)^{-1} B_{n\times m}\right]$ to obtain $M_{r\times n} = T_{r\times m}C_{m\times n}$.

Proposed Digital Signature Scheme

a. Key Generation Algorithm

The sender chooses an arbitrary matrix $B_{n\times m}, C_{m\times n}$, where $m > n$ is a rectangular (singular matrix) matrix, and computes $A_{n\times n} = B_{n\times m}C_{m\times n}$.

Next, the sender computes $\left(A_{n\times n}\right)^{-1}$. Repeat the first step if it is not found. If $\left(A_{n\times n}\right)^{-1}$ exists, the sender calculates the private key $E_{n\times m} = \left(A_{n\times n}\right)^{-1} B_{n\times m}$ and prepares the public key $C_{m\times n}$.

b. Signatory

Suppose the sender of a message has a message $M_{r\times n}$. The sender of the message signs the message by computing $T_{r\times m} = M_{r\times n}E_{n\times m}$.

c. Signature Verification

The recipient verifies the message's authenticity by computing $M_{r\times n} = T_{r\times m}C_{m\times n}$, where $C_{m\times n}$ is the public key.

We can show that signature verification can work well, namely,

$$\begin{aligned} T_{r\times m}C_{m\times n} &= M_{r\times n}E_{n\times m}C_{m\times n} \\ &= M_{r\times n}\left(A_{n\times n}\right)^{-1} B_{n\times m}C_{m\times n} \\ &= M_{r\times n}\left(A_{n\times n}\right)^{-1} A_{n\times n} \\ &= M_{r\times n} \end{aligned}$$

A Toy Example

The sender has a message "TAX". Suppose the sender and receiver agree to use the terms $A = 01, B = 02, \ldots, Z = 0$. The sender uses the private key of the matrix 2×2. The message becomes "TAX" and is changed to "20-01-24". A matrix of size 1×2 is formed, namely $M_{1\times 2} = \begin{bmatrix} 200 & 124 \end{bmatrix}$, the term $r < n$. In this example, any prime modulo $p = 1231$ is chosen.

a. Key Generation Algorithm

The message's sender chooses any matrix $B_{2\times 3} = \begin{bmatrix} 11 & 101 & 512 \\ 33 & 201 & 701 \end{bmatrix}$,

$C_{3\times 2}=\begin{bmatrix}311 & 17\\ 11 & 122\\ 101 & 1111\end{bmatrix}$. The sender computes $A_{2\times 2}=B_{2\times 3}C_{3\times 2}=\begin{bmatrix}849 & 309\\ 798 & 51\end{bmatrix}\bmod 1231$.

Note that $\left(A_{2\times 2}\right)^{-1}=\begin{bmatrix}857 & 1035\\ 928 & 1160\end{bmatrix}\bmod 1231$ exists, so the message sender computes the private key $E_{2\times 3}=\begin{bmatrix}497 & 383 & 1024\\ 479 & 673 & 670\end{bmatrix}\bmod 1231$ and the public key $C_{3\times 2}=\begin{bmatrix}311 & 17\\ 11 & 122\\ 101 & 1111\end{bmatrix}$.

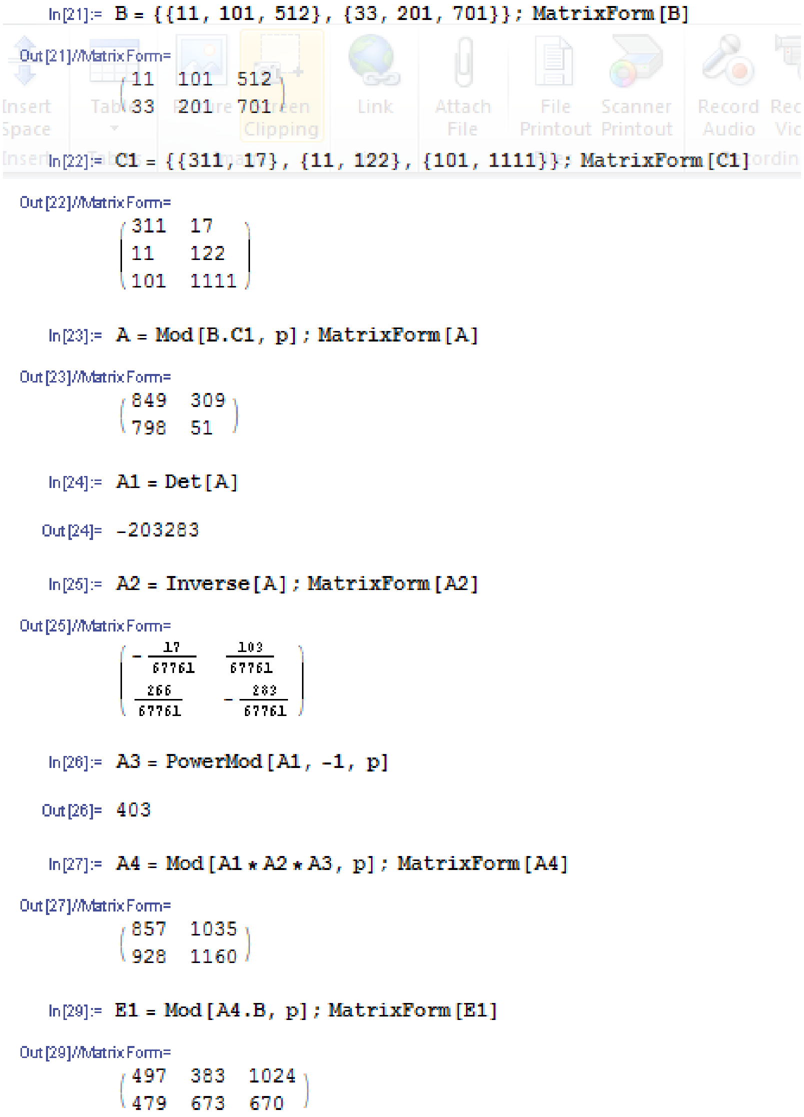
b. Signatory

Suppose the sender of a message has a message $M_{1\times 2}=\begin{bmatrix}200 & 124\end{bmatrix}$. The sender of the message signs the message by computing $T_{1\times 3}=\begin{bmatrix}1228 & 22 & 1057\end{bmatrix}\bmod 1231$.

c. Signature Verification

The message's recipient verifies its authenticity by computing $M_{1\times 2}=T_{1\times 3}C_{3\times 2}=\begin{bmatrix}200 & 124\end{bmatrix}$, where $C_{m\times n}$ is the public key. Thus, the message is unchanged. The recipient can translate the message back to "20-01-24" or "TAX".

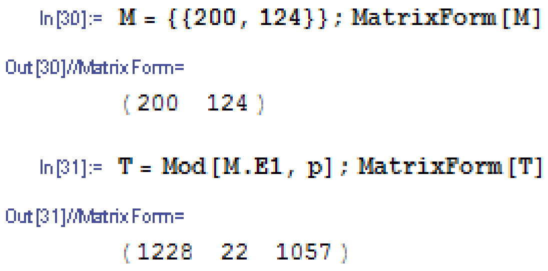
The following is a display of the algorithm using Mathematica software.



```
In[21]:= B = {{11, 101, 512}, {33, 201, 701}}; MatrixForm[B]

Out[21]//MatrixForm=
         ( 11   101  512 )
         ( 33   201  701 )

In[22]:= C1 = {{311, 17}, {11, 122}, {101, 1111}}; MatrixForm[C1]

Out[22]//MatrixForm=
         ( 311  17   )
         ( 11   122  )
         ( 101  1111 )

In[23]:= A = Mod[B.C1, p]; MatrixForm[A]

Out[23]//MatrixForm=
         ( 849  309 )
         ( 798  51  )

In[24]:= A1 = Det[A]

Out[24]= -203283

In[25]:= A2 = Inverse[A]; MatrixForm[A2]

Out[25]//MatrixForm=
         ( -17/67761    103/67761 )
         ( 266/67761   -283/67761 )

In[26]:= A3 = PowerMod[A1, -1, p]

Out[26]= 403

In[27]:= A4 = Mod[A1 * A2 * A3, p]; MatrixForm[A4]

Out[27]//MatrixForm=
         ( 857  1035 )
         ( 928  1160 )

In[29]:= E1 = Mod[A4.B, p]; MatrixForm[E1]

Out[29]//MatrixForm=
         ( 497  383  1024 )
         ( 479  673  670  )
```

Figure 2. Key Generation Simulation in Mathematica

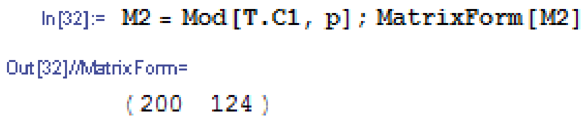
Next, the signing process is continued by the following algorithm.



```
In[30]:= M = {{200, 124}}; MatrixForm[M]

Out[30]//MatrixForm=
        ( 200   124 )

In[31]:= T = Mod[M.E1, p]; MatrixForm[T]

Out[31]//MatrixForm=
        ( 1228   22   1057 )
```

Figure 3. Signatory in Mathematica

At the end, the message recipient will verify the signature so that the recipient is sure of the message's authenticity.



```
In[32]:= M2 = Mod[T.C1, p]; MatrixForm[M2]

Out[32]//MatrixForm=
        ( 200   124 )
```

Figure 4. Signature Verification in Mathematica

Mathematical Analysis of Scheme Security

a. Key generation Algorithm

The sender generates a private key $E_{n\times m}=\left(A_{n\times n}\right)^{-1}B_{n\times m}$ and a public key $C_{m\times n}$. In this condition, the matrix E,A,B can only be brute-forced. This condition is safe because extra effort is required to guess any matrix E,A,B correctly.

b. Signatory

In the signing phase, the sender sends a message $M_{r\times n}$. The sender signs the message by calculating $T_{r\times m}=M_{r\times n}E_{n\times m}$. It is clear that $M_{r\times n}$ and $T_{r\times n}$ are not secret (because they will be sent to the message's recipient), so the eavesdropper can do $E_{n\times m}=\left(M_{r\times n}\right)^{-1}T_{r\times m}$ to steal the private key. But this condition is impossible because $\left(M_{r\times n}\right)^{-1}$ does not exist ($M_{r\times n}$, with $r<n$, is a singular rectangular matrix that does not have an inverse).

Furthermore, the hacker's $T_{r\times m}=M_{r\times n}E_{n\times m}$ equation can be tried to be tapped by multiplying any matrix $K_{n\times r}$ from the left side of M, namely

$$K_{n\times r}T_{r\times m}=K_{n\times r}M_{r\times n}E_{n\times m}$$

$$\left(KT\right)_{n\times m}=\left(KM\right)_{n\times n}E_{n\times m}$$

The goal is to form a square matrix $\left(KM\right)_{n\times n}$ so that $\left(KM\right)_{n\times n}^{-1}\left(KT\right)_{n\times m}=E_{n\times m}$ applies.

The hacker still wants to steal the private key, but this condition is impossible because $\left(KM\right)_{n\times n}^{-1}$ does not exist. This is supported by mathematical theory (algebra) that $K_{n\times r}M_{r\times n}$ is a singular matrix (a matrix that does not have an inverse/singular matrix) because of the condition $r<u$ (Maxrizal & Irawadi, 2020).

Therefore, during the signing phase, the private key E is impervious to mathematical attacks. This is why the size of the message matrix M is predetermined, preventing this type of attack by hackers. The key matrix is only susceptible to brute force attacks.

c. Signature Verification

In the signature verification phase, all variables are not confidential. In this condition, it is not profitable for hackers to perform brute force attacks.

Comparison with Previous Scheme

In this study, we compare the proposed scheme with the previous scheme (Gupta & Sanghi, 2021) in terms of mathematical theory.

Table 1. Comparison with Previous Scheme

No.	Criteria being Compared	Previous Schemes	Proposed Scheme
1.	Key generation process	Matrix	Matrix
2.	Key generator	Generated from the multiplication of prime numbers $n = pq$.	Any s
3.	Private key form	Integer	Matrix
4.	Signature computation speed	Involves raising integers to powers modulo n of relatively large magnitude.	Involves ordinary matrix multiplication
5.	Possible brute force attack on private key	n_1 possibility	$E_{n\times m}$ on $\bmod n_1$ with $m > n$ then there are T n_1^{nm} possibilities

The above checks are carried out on the variables involved in the key formation, signing, and signature verification processes.

CONCLUSIONS AND SUGGESTIONS

This study proposes a digital signature scheme using the matrix concept. The results show that the proposed scheme can generate keys and sign and verify signatures well. In addition, the proposed scheme system has also been tested for possible mathematical attacks.

Further research can examine the algorithm's performance time on key generation, signature, and signature verification. In addition, further research can also see attacks from the quantum attack algorithm's side.

ACKNOWLEDGEMENT

This research was funded by the Ministry of Education, Culture, Research, and Technology through the Beginner Lecturer Research Grant (PDP) scheme with contract number 104/E5/PG.02.00/PL/2024. For that, the researcher would like to thank the Ministry of Education, Culture, Research, and Technology for providing the budget for this research. The author would also like to thank the Atma Luhur Institute of Science and Business for holding many training activities and research writing workshops so that the author could obtain this grant.

REFERENCES

Anshori, Y., Dodu, A. Y. E., & Wedananta, D. M. P. (2019). Implementasi algoritma kriptografi rivest shamir adleman (rsa) pada tanda tangan digital. *Techno.Com*, *18*(2), 110–121. https://doi.org/10.33633/tc.v18i2.2166

Fuchsbauer, G., Plouviez, A., & Seurin, Y. (2020). Blind schnorr signatures and signed elgamal encryption in the algebraic group model. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in*

Bioinformatics), 12106 LNCS. https://doi.org/10.1007/978-3-030-45724-2_3

Gupta, S. C., & Sanghi, M. (2021). Matrix modification of rsa digital signature scheme. *Journal of Applied Security Research*, *16*(1), 63–70. https://doi.org/10.1080/19361610.2019.1690350

Ismail, N. H. M., & Misro, M. Y. (2022). Bézier coefficients matrix for elgamal elliptic curve cryptosystem. *Malaysian Journal of Mathematical Sciences*, *16*(3), 483–499. https://doi.org/10.47836/mjms.16.3.06

Lalem, F., Laouid, A., Kara, M., Al-Khalidi, M., & Eleyan, A. (2023). A novel digital signature scheme for advanced asymmetric encryption techniques. *Applied Sciences*, *13*(8), 5172. https://doi.org/10.3390/app13085172

Maxrizal, M., & Irawadi, S. (2019). Modifikasi protokol tanda tangan digital elgamal menggunakan general linear group. *Jurnal Matematika Integratif*, *15*(1), 39. https://doi.org/10.24198/jmi.v15.n1.20960.39-44

Maxrizal, M., & Irawadi, S. (2020). Analisis sistem kriptografi elgamal untuk membentuk sistem kunci publik berbasis grup non-komutatif. *Jurnal Matematika Integratif*, *16*(2), 117. https://doi.org/10.24198/jmi.v16.n2.29197.117-125

Qin, Y., & Zhang, B. (2023). Privacy-preserving biometrics image encryption and digital signature technique using arnold and elgamal. *Applied Sciences*, *13*(14), 8117. https://doi.org/10.3390/app13148117

Raj, B. S. S., & Sridhar, V. (2021). Identity based cryptography using matrices. *Wireless Personal Communications*, *120*(2), 1637–1657. https://doi.org/10.1007/s11277-021-08526-9

Saepulrohman, A., & Negara, T. P. (2021). Implementasi algoritma tanda tangan digital berbasis kriptografi kurva eliptik diffie-hellman. *Komputasi: Jurnal Ilmiah Ilmu Komputer Dan Matematika*, *18*(1), 22–28. https://doi.org/10.33751/komputasi.v18i1.2569

Saputra, R. A., & Purnomo, A. S. (2018). Implementasi algoritma schnorr untuk tanda tangan digital. *JMAI (Jurnal Multimedia & Artificial Intelligence)*, *2*(1), 21–26. https://doi.org/10.26486/jmai.v2i1.69

Swain, S. M., Pradhan, A., & Moharana, S. K. (2022). A comparative study on digital signature schemes. In *International Journal of Current Science* (Vol. 12).